IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: GARY STEPHEN SHUSTER

Serial No.: 09/837,319

Filed: April 18, 2001

Title: METHOD AND SYSTEM FOR OPERATING A NETWORK SERVER TO DISCOURAGE INAPPROPRIATE USE

Art Unit: 2145

Examiner: Azizul Q. Choudhury

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir or Madam:

The appellant filed a Notice of Appeal in the above-identified application on August 10, 2007 under 35 U.S.C. § 134(a). The Appeal Brief meets the substantive requirements of Rule 41.37. The appellant requests entry, consideration, and favorable action on this appeal at the Office's earliest convenience.

In accordance with Rule 41.37(c), the appellant presents the following items under the headings prescribed therein.

Real Party in Interest

Hoshiko, LLC, a Nevada Limited Liability Company, owns the subject application.

Related Appeals and Interferences

None.

Status of Claims

As stated in the Official Action mailed on May 11, 2007 (hereinafter the "Final Action"), claims 1-20 stand rejected. On August 10, 2007, the appellant filed a Notice of Appeal from the rejections of pending Claims 1-20.

Status of Amendments

Claims 1 and 11 were amended subsequent to the Final Action. These amendments were entered as noted in the advisory action mailed on August 1, 2007.

Summary of Claimed Subject Matter

This section includes a concise explanation of the subject matter defined in each of the independent claims involved in the appeal (i.e., claims 1 and 11), which includes references to the specification and drawings and other information as specified in Rule 41.37.

Claim 1 defines a method for operating a network server to discourage use that disproportionately depletes server resources such as distribution of large media files, wherein the server is connected to a plurality of client devices, and configured to transfer information between selected ones of the client devices and a memory for static storage of information. See page 5, lines 6-17. The method comprises the steps of:

(A) "[*R]eceiving a request to transfer a file between the memory and one of the plurality of client devices*." Page 5, lines 12-13; fig. 2 at 21. The network server receives a request for a file to be transferred to a client device.

(B) "*[R]emoving a packet of information from the file after said receiving step.*" Page 6, lines 4-17; fig. 2 at 23. The network server removes a packet of defined size for transfer to the client. The packet may be removed by copying a defined portion of the file into a working memory in the server and recording information for later reassembly of the file. Page 6, lines 9-12.

(C) "*[T]ransferring the packet of information from the memory to a lower-level network component operative to configure the packet as at least one lower-level packet according to a protocol of a packet-switched network for transmission to the one of the plurality of client devices after said removing step.*" Page 6, lines 18-29; fig. 2 at 24; page 5, line 29 – page 6, line 3. After the packet is removed, it is transferred to the client device via a packet-switched network. The high-level packet used for control of transmission rate is therefore configured as a lower-level packet for transmission over the network.

(D) "*[P]ausing for a defined delay period after at least one of said removing and said transferring steps.*" Page 7, lines 1-6; fig. 2 at 26. A pause is introduced for a defined delay period.

(E) *"[R]epeating said removing step, said transferring step, and said pausing step in any operative order until all of the file has been transferred to the lower-level network component, wherein at least one of the delay period and the a defined number of information bits in the information packet is controlled so as to cause later-transferred*

portions of the file to be delayed by increasing amounts until all portions of the file have been transferred, whereby the entire file is transferred at a rate that decreases with increasing file size." Page 8, lines 19-22; fig. 2 at 23-28; page 9, line 10 – page 10, line 9. Periodically, a pause is inserted in the packetization/transfer process having the effect of decreasing the transfer rate with increasing file size. Thus, for example, there will be imparted "an unnoticeable additional delay of 45 milliseconds in transferring a 100 kilobyte file; a slight additional delay of 4.95 seconds in transferring a one megabyte file; a substantial additional delay of 8 minutes and 19 seconds in transferring a ten megabyte file, and a probably intolerable additional delay of 208 hours in transferring a fifty megabyte file." Page 9, line 28 – page 10, line 3.

Claim 11 defines a system for discouraging use of memory connected to a network where the use, such as distribution of large media files, disproportionately depletes server resources. The system comprises:

(A) *"[A] memory for static storage of information*." Page 4, lines 13-17; fig. 1 at 18. A server has a memory device for static storage of information; examples include a hard drive or optical disk drive.

(B) "*[A] server connected to a plurality of client devices and to the memory, the server controlling access by the client devices to the memory.*" Page 4, lines 4-14; fig. 1 at 16. A server may comprise a general purpose computer configured for serving information to multiple users across a network.

(C) "*[A]n application on the server for transferring information between selected ones of the client devices and the memory, the application comprising instructions to perform the steps of:*

receiving a request to transfer a file between the memory and one of the plurality of client devices;

removing a packet of information from the file after the receiving step;

transferring the packet of information from the memory to a lower-level network component operative to configure the packet as at least one lower-level packet according to a protocol of a packet-switched network for transmission to the one of the plurality of client devices after the removing step;

pausing for a defined delay period after one of the removing and the transferring steps;

repeating the removing step, the transferring step, and the pausing step in any operative order until all of the file has been transferred to the lower-level network component, wherein at least one of the delay period and the a defined number of information bits in the information packet is controlled so as to cause later-transferred portions of the file to be delayed by increasing amounts." Page 4, lines 4-14; fig. 1 at 18. An application resides on the server for performing the method claimed. The application comprises instructions for carrying out the steps in the same way as defined

for claim 1. Accordingly, the explanation of this subject matter is the same as provided above in connection with claim 1; see above.

Grounds of Rejection To Be Reviewed on Appeal

Claims 1-20 stand rejected under 35 § 102(b) over Kalkunte et al. (U.S. Pat. No. 5,854,900). This ground of rejection is to be reviewed on appeal. No other grounds for rejection have been set forth.

Argument

In the arguments below, the appellants present reasons why defined groups of claims are separately patentable over the cited references. Claims that are not separately discussed stand or fall with their respective base claims.

MPEP § 2131 states the basic requirements for anticipation under § 102 as follows:

> "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

The present rejections rely on a reference, U.S. Pat. No. 5,854,900 "Kalkunte," that does not disclose each and every element as set forth in the independent claims and in certain independent claims, as set forth more fully below. Therefore, Kalkunte does not anticipate the claims under § 102, which are therefore allowable.

A. Claims 1 and 11

Kalkunte fails to disclose all the limitations of independent claims 1 and 11, and therefore cannot anticipate these claims.

Kalkunte fails to disclose:

> repeating the removing step, the transferring step, and the pausing step in any operative order until all of the file has been transferred to the lower-level network component, wherein at least one of the delay period and a defined number of information bits in the information packet is controlled so as to cause later-transferred portions of the file to be delayed by increasing amounts until all portions of the file have been transferred, whereby the entire file is transferred at a rate that decreases with increasing file size

as defined by claims 1 and 11. Kalkunte fails to disclose or suggest several aspects of the recited subject matter.

In general, contrary to the objectives described by the present application and claims, Kalkunte is concerned with entirely different objectives of increasing network throughput and preventing traffic collisions. Col. 2:62-6; col. 3:1-13. At most, Kalkunte discloses that under some circumstances (for example, if a collision is detected) a packet may be delayed so as to avoid network collisions. *See, e.g.,* col. 5:54-57. Under other circumstances during transmission of the same file, Kalkunte teaches reducing the delay interval to zero to accomplish a faster overall transmission. Col. 3:35-46; 6:45-55. Far from disclosing every feature of claims 1 and 11, Kalkunte actually teaches away from the claimed combination, when considered fairly as a whole. According to the claims 1 and 11, the packets are controlled so as to cause

transmission delays that increase with increasing file size, by increasing delays for later portions of a file. In contrast, Kalkunte teaches how to *increase* network throughput, which necessarily requires *decreasing* transmission times and conversely *increasing* transmission rates. Col. 2:62-64; 8:21-29; Tables 1-3. Several deficiencies of Kalkunte with respect to the cited claim language are considered in turn below.

1. "Whereby The Entire File Is Transferred At A Rate That Decreases With Increasing File Size"

Kalkunte fails to disclose or suggest increasing the delay interval or reducing the packet size so as to cause the entire file to be transferred at a rate that decreases with increasing file size, as defined by claims 1 and 11.

It was argued in the last Final Action that the claimed result of "whereby the entire file is transferred at a rate that decreases with increasing file size" is inherent in file transfer as disclosed by Kalkunte. This is not so. Inherency may only be found when the claimed feature is *necessarily present* in the thing described by prior-art reference. M.P.E.P. § 2163.07(a). Specifically, it was argued in the Final Action that "[w]hen more packets are sent without changing the bandwidth or transfer medium, it is inherent that their transfer rate will decrease." Final Action, page 9. This statement is not supported and is somewhat unclear; it appears to be a statement that as network traffic increases on a fixed pipe, transmission rates per file decrease. It cannot mean that larger files must transfer at a slower data rate that small files, when bandwidth is

the same. It is self apparent, for example, that one may transfer a 1 MB file and a 10 MB file at the same rate (e.g., 100 MB/second), or at different rates, through a given transfer medium. If at different rates, which file will transfer at the slower rate is not predictable based on file size. On the other hand, if the statement means that as network traffic increases on a fixed pipe, transmission rates per file decrease, it may be more plausible, although still unsupported. However, such an argument would not be relevant to anticipation of claims 1 and 11, because these claims do not define or require an increase in traffic.

Instead, claims 1 and 11 require that "at least one of the delay period and a defined size of the information packet is controlled so as to cause later-transferred portions of the file to be delayed by increasing amounts, whereby the entire file is transferred at a rate that decreases with increasing file size." In network media generally, transfer rates may go up or down with the number of packets being transmitted at a given moment of time, that is, the amount of network traffic, and other network conditions. However, as defined by claims 1 and 11, the packets that are used to transmit a file are *not* in transit at the same time. As claimed, an increasing delay period is inserted between the packets. These claims therefore define a situation in which the packets making up a file are transmitted sequentially, because of the iterative steps and further because of the insertion of pauses between transfer steps. Therefore, no matter how many packets make up the file, any increase in network traffic caused by

transmission of the file is unrelated to the file size, because no more than one packet is being transmitted at any given time. In this circumstance, transmission rates are inherently not related to file size in prior-art Ethernet networks such as disclosed by Kalkunte.

Of course, the Final Action necessarily relies on a conclusion that Kalkunte discloses the same type of sequential packet transmission as defined by claims 1 and 11. If otherwise, there could be no anticipation of these claims. We have just shown, however, that in packeted sequential transmissions, there is no inherent relationship between file size and transmission rate. It should therefore be clear that the claimed relationship between file size and transmission rate cannot be inherent to the disclosure of Kalkunte. At the very least, the conclusion of inherency on which the Final Action's finding of anticipation under §102(b) depends has not been justified either by a reasonable argument or by evidence.

If the statement in the Final Action means that it is inherent that larger files require more time to transfer than smaller files, this too misses the mark. Given a constant bandwidth, it is a mere truism that the larger the file, the more time required to transfer it. However, claims 1 and 11 cannot reasonably be read as reciting this mere truism. Both claims require that the *rate* at which the file transfers decrease with increasing file size. "Rate" means amount of data transferred per unit of time, for example, mega-Bytes per second. This is different from merely the increasing the

transfer *time* with increasing file size. While it is apparent that file size inherently impacts transfer time given a set bandwidth, file size does not necessarily or even ordinarily affect transfer *rate*. Claims 1 and 11 define that causing "later-transferred portions of the file to be delayed by increasing amounts, *whereby* the entire file is transferred at a rate that decreases with increasing file size." Therefore, the claims define a circumstance in which the increase in transfer rate with file size is caused by an increasing delay in *later portions* of the file. This cause/effect is not disclosed by the neutral packetization scheme of Kalkunte, in which delay is used to avoid packet collision, thereby increasing efficiency and throughput. Far from disclosing this element of claims 1 and 11, Kalkunte teaches away from it.

2. "So As To Cause Later-Transferred Portions Of The File To Be Delayed By Increasing Amounts"

Nor has reasonable consideration been given to the claimed feature of controlling the packet size or delay period "so as to cause later-transferred portions of the file to be delayed by increasing amounts." Such a feature is not inherent in Kalkunte or any prior-art network.

The Final Action erroneously cites col. 3:9-61 as disclosing this feature, but Kalkunte nowhere discloses controlling the packet size or delay period "so as to cause later-transferred portions of the file to be delayed by increasing amounts." This feature is most reasonably construed to encompass a situation in which later-transferred

portions of a file are delayed for progressively longer periods of time – this is the broadest reasonable meaning of "delayed by increasing amounts." Therefore the claims define a method in which the amount of delay depends on which portion (i.e., earlier or later) of the file is being transferred.

Kalkunte fails to disclose such a feature. To the contrary, Kalkunte discloses two embodiments, neither of which include the claimed feature. For its first embodiment, Kalkunte teaches "waiting a delay time that includes a predetermined interpacket gap interval and one slot time if the detecting step detects the second data packet." Col. 3:19-22. The second data packet refers to a second packet waiting to be transmitted. Col. 3:17-19.

"Slot time" is used in Kalkunte consistent with its meaning in the art of Ethernet transmissions, to denote the amount of time a device waits after a collision before retransmitting. The rationale and effect of this method is described in Kalkunte at col. 6:17-29 as a way to avoid collisions on 2-station and 3-station Ethernet networks. The duration of the slot time is according to convention 512 bit times for 10 and 100 Mbit/s networks, and preferably 4096 bit times for 1000 Mbit/s networks. Col. 6:2-10. The slot time is either constant, or for 1000 Mbit/s networks only is "selected in accordance with the network topology and propagation delay." Either way, this first Kalkunte embodiment fails to disclose selecting or controlling a slot time "so as to cause later-transferred portions of the file to be delayed by increasing amounts." Again, Kalkunte is

inimical to increasing amounts of delay for later-transferred file portions, seeking only to avoid network collisions and any resulting transmission delay.

For its second embodiment, Kalkunte teaches calculating, when a network collision occurs, a "collision delay interval . . . as an integer multiple of a predetermined slot time randomly selected from a range of intervals *calculated from an exponential number of the access attempts*." Col. 3:46-50, emphasis added. So if a collision occurs, the delay interval is based on network conditions, not on what portion of the file is being transferred. If no collision occurs, the delay interval is set to zero. Col. 3:30-45. According to Kalkunte, the delay interval is determined is based on network conditions, completely without reference to or effect on which portion of the file is being transferred. Indeed, setting the delay interval to zero could cause later-transferred portions of the file to be transferred by *decreasing* amounts, opposite to what is claimed.

In both of its embodiments, Kalkunte does not consider the file size or what portion of a file is being transmitted to calculate a delay interval. Kalkunte therefore fails to disclose controlling the delay interval or packet size so as to cause later portions of the file to be delayed by increasing amounts.

3. High-level Packetization

In addition, Kalkunte fails to disclose or suggest forming packets at a higher level, and then transferring the packet to a lower-level component for packeting according to a network protocol. That is, Kalkunte fails to disclose or suggest:

> removing a packet of information from the file after the receiving step;
>
> transferring the packet of information from the memory to a lower-level network component operative to configure the packet as at least one lower-level packet according to a protocol of a packet-switched network for transmission to the one of the plurality of client devices after the removing step,

wherein the packets of information are handled as defined elsewhere by claims 1 and 11.

In the Final Action at page 3, Kalkunte is cited at col. 4:49-57 as disclosing the claim language recited above. Kalkunte there discloses passing information byte-by-byte from a PCI bus interface to a transmit FIFO register of an Ethernet controller. Transferring bytes to a FIFO register as part of a caching operation for the MAC core fails to teach or suggest the claimed feature, because, among other things, a “byte” is not a “packet” as these claims define. The present specification describes a packet as “preferably the same or larger than the size of the typical packet size of the transmission control protocol in use on the network over which the file will be transferred, such as, for example, 8192 (8k) bytes.” Page 5:27-29. A single byte is one of the lowest-level group of information bits used in computing, and it is not reasonable to construe a “byte” as a “packet” similar in size to a packet in a packet-switched network. The claim language requires that the packet be transferred to “a lower-level network component operative to configure the packet as at least one lower-level packet according to a protocol of a packet-switched network.” Because the packet is configured as a lower-level packet, it is therefore a higher level packet than the lower-level packet that is

configured according to the protocol of the packet-switched network. In contrast, Kalkunte teaches that the information is transferred on a *byte-by-byte* basis from the PCI bus interface to the FIFO register of an Ethernet component. Col. 4:50-57. Kalkunte, therefore, teaches passing data to the FIFO register in groups of one byte each. As is well known in the art, an Ethernet packet includes multiple bytes of data and address information; it cannot be reasonably construed as a "lower-level packet" as compared to a byte. Therefore, in disclosing transfer of bytes from a PCI bus interface to an Ethernet controller, Kalkunte fails to disclose the recited steps.

Likewise, transferring data via a PCI bus from a CPU to a PCI bus interface does not read on the claim language, because the PCI bus is not "a lower-level network component operative to configure the packet as at least one lower-level packet according to a protocol of a packet-switched network." A PCI bus would be understood as servicing an internal bus, not a packet-switched network. Here too Kalkunte fails to disclose the recited steps.

If the PCI bus interface is ignored, in effect Kalkunte discloses a transfer of data from a CPU to a FIFO register of an Ethernet controller. As such, however, Kalkunte fails to disclose or suggest the removing of the transferred packet from a file. Kalkunte fails to disclose or suggest packetizing files before transferring via a local bus. No matter how the transfer of a file from a CPU to the Ethernet controller is analyzed, it is not the same as these recited steps of claims 1 and 11.

In summary, Kalkunte fails to disclose (a) repeating the removing step, the transferring step, and the pausing step . . . whereby the entire file is transferred at a rate that decreases with increasing file size, (b) repeating removing, transferring and pausing steps . . . so as to cause later-transferred portions of the file to be delayed by increasing amounts, and (c) forming packets at a higher level, and then transferring the packet to a lower-level component for packeting according to a network protocol. Failing to disclose all the limitations of claims 1 and 11, Kalkunte cannot anticipate these claims, which are therefore allowable. All of the remaining claims are also allowable, at least as depending from one of these allowable base claims.

B. Claims 2 and 12

Claims 2 and 12 are allowable as depending from allowable base claims, and are also independently allowable over Kalkunte. Kalkunte fails to disclose or suggest increasing the defined delay period after each iteration of the repeating step, as defined by claims 2 and 12. To the contrary, Kalkunte teaches away from this, by teaching that the delay period should be reset to zero to give priority to "the station having deferred once to another station." 6:45-55.

In the Final Action, it was argued that Kalkunte teaches that "delay intervals are adjustable and delays can be increased." Final Action, page 9. Even if true, this is irrelevant because Kalkunte fails to disclose or suggest the recited feature of increasing the delay interval *after each iteration*. Increasing the delay interval after each iteration is

plainly contrary to Kalkunte's stated purpose of increasing network throughput. Col. 2:62-64. Given a constant bandwidth and transfer medium, throughput can only be increased by increasing transmission rate. Kalkunte teaches increasing delays only to prevent collisions and the longer delays that collisions would cause. Col. 3:23-30. Far from teaching the claimed feature, Kalkunte teaches away from it.

Therefore, failing to disclose or suggest all the claimed elements of Claims 2 and 12, Kalkunte presents no bar to patentability under 35 U.S.C. § 102.

C. Claims 7 and 17

In addition to being allowable as depending from allowable base claims, claims 7 and 17 are independently allowable, because Kalkunte fails to disclose or suggest determining the calculated value for a delay period from the file size or the file type so as to cause the entire file to be transferred at a rate that decreases with increasing file size, as defined by claims 7 and 17. Kalkunte discloses modifying delay times based only on network conditions, not on file size. In every instance disclosed in Kalkunte, the delay interval is determined based on network conditions. Col. 5:23 – 7:49; figs. 3A-3C.

In the Final Action it was argued only that Kalkunte discloses setting a delay value based on network factors, including packet size. Packet size is not synonymous with file size under any reasonable claim construction. A frame passed to a FIFO register, as disclosed in Kalkunte at 4:61-63, lacks the high-level characteristic of a file, such as a file name. Moreover, a data frame as disclosed by Kalkunte cannot read

simultaneously on both “packet” and “file,” which are separate claim terms used for different and distinct objects.

In summary, Kalkunte is concerned only with whether or not a collision has occurred in determining a delay period. Col. 5:23 – 7:49; Figs. 3A-3C. Kalkunte fails to disclose or suggest “determining the calculated value from the file size or the file type,” as defined by claims 7 and 17. These claims are therefore independently allowable.

Therefore, failing to disclose or suggest all the claimed elements of claims 7 and 17, Kalkunte presents no bar to patentability under 35 U.S.C. § 102.

D. Claims 8 and 18

In addition to being allowable as depending from allowable base claims, claims 8 and 18 are independently allowable, because Kalkunte fails to disclose or suggest “setting the defined number of information bits in the packet of information to a calculated value after each execution of the pausing step.” This requires that the packet size be determined by a value that is calculated, as opposed to being constant or pre-determined. In addition, the packet size must be set after each iteration, which rules out use of a constant packet size. It would be unreasonable to hold that a network using a fixed packet size during a given transmission is actually “setting” the packet size “to a calculated value” after each packet is transmitted. To so hold would rob what is defined by claims 1 and 18 of any discernable meaning. Yet such a clearly erroneous holding is what the Final Action relies on, and nothing more.

Kalkunte nowhere discloses changing the number of information bits in a packet at any time. To the contrary, it is unrebutted that Kalkunte says nothing about packet size. This silence is alone enough to obviate the alleged case of anticipation. The claimed feature is simply not disclosed.

Nor can Kalkunte be reasonably held to inherently disclose the claimed subject matter. Because of its silence about packet size, Kalkunte could easily be understood as using a fixed packet size for a given transmission or network type. It is respectfully submitted that networks do not normally set packet size to a calculated value after each packet is transmitted, if ever. No reference has been provided for the statement on page 5, para. 8 of the Final Action that "[n]etworks allow the size of packets to be set as claimed." This argument is utterly circular and unsupported, and should be given no weight. Nor has Official Notice been taken of any aspect of network operation that would demonstrate any inherent disclosure by Kalkunte of the claimed subject matter. That a network might permit setting of packet size during transmission, even if true (which is not admitted), cannot demonstrate that setting of packet size to a calculated value after every iteration would inherently be present in what is disclosed by Kalkunte. Demonstrating a mere possibility does not amount to proving that the claimed feature is *necessarily present*, as required for a showing of anticipation. In the absence of a single reference showing all the claimed subject matter, including setting of packet size to a calculated value after every iteration of the pausing step, claims 1 and 18 are

independently allowable.

Therefore, failing to disclose or suggest all the claimed elements of claims 8 and 18, Kalkunte presents no bar to patentability under 35 U.S.C. § 102.

Conclusion

Appellants respectfully request the reversal of the rejection of currently pending Claims 1-20, and allowance of these claims forthwith, for the reasons set forth above.

Appendix

Appealed Claims 1-20 are attached hereto as Appendix A. Evidence for consideration in this appeal is attached hereto as Appendix B. Related Appeals and Interferences, if any, are listed in Appendix C.

Respectfully submitted,

Date: October 10, 2007

/Jonathan Jaech/

Jonathan Jaech
Attorney for Applicant
Registration No. 41,091

CUSTOMER NUMBER

58688

PATENT TRADEMARK OFFICE

Connolly Bove Lodge & Hutz LLP
355 South Grand Avenue
Suite 3150
Los Angeles, CA 90071-1560
(213) 787-2500

APPENDIX A

APPEALED CLAIMS

1. (Previously presented) A method for operating a network server to discourage use that disproportionately depletes server resources such as distribution of large media files, wherein the server is connected to a plurality of client devices, and configured to transfer information between selected ones of the client devices and a memory for static storage of information, said method comprising the steps of:

receiving a request to transfer a file between the memory and one of the plurality of client devices;

removing a packet of information from the file after said receiving step;

transferring the packet of information from the memory to a lower-level network component operative to configure the packet as at least one lower-level packet according to a protocol of a packet-switched network for transmission to the one of the plurality of client devices after said removing step;

pausing for a defined delay period after at least one of said removing and said transferring steps; and

repeating said removing step, said transferring step, and said pausing step in any operative order until all of the file has been transferred to the lower-level network component, wherein at least one of the delay period and a defined number of information bits in the information packet is controlled so as to cause later-transferred portions of the file to be delayed by increasing amounts until all portions of the file have been transferred, whereby the entire file is transferred at a rate that decreases with increasing file size.

2. (Previously presented) The method of Claim 1, further comprising increasing the defined delay period after each iteration of the repeating step.

3. (Original) The method of Claim 1, further comprising setting the defined delay period to a selected predetermined value after each execution of said pausing step.

4. (Original) The method of Claim 1, further comprising initializing the defined delay period to a calculated value prior to said removing step.

5. (Original) The method of Claim 1, further comprising initializing the defined delay period to a selected predetermined value prior to said removing step.

6. (Original) The method of Claim 1, further comprising setting the defined delay period to a calculated value after each execution of said pausing step.

7. (Previously presented) The method of Claim 6, further comprising determining the calculated value from the file size or file type.

8. (Original) The method of Claim 1, further comprising setting the defined number of information bits in the packet of information to a calculated value after each execution of said pausing step.

9. (Original) The method of Claim 1, further comprising setting the defined number of information bits in the packet of information to a selected predetermined value after each execution of said pausing step.

10. (Original) The method of Claim 1, further comprising initializing the defined number of information bits in the packet of information prior to said removing step.

11. (Previously presented) A system for discouraging use of memory connected to a network where the use, such as distribution of large media files, disproportionately depletes server resources, the system comprising:

a memory for static storage of information;

a server connected to a plurality of client devices and to the memory, the server controlling access by the client devices to the memory; and

an application on the server for transferring information between selected ones of the client devices and the memory, the application comprising instructions to perform the steps of:

receiving a request to transfer a file between the memory and one of the plurality of client devices;

removing a packet of information from the file after the receiving step;

transferring the packet of information from the memory to a lower-level network component operative to configure the packet as at least one lower-level packet according to a protocol of a packet-switched network for transmission to the one of the plurality of client devices after the removing step;

pausing for a defined delay period after one of the removing and the transferring steps;

repeating the removing step, the transferring step, and the pausing step in any operative order until all of the file has been transferred to the lower-level network component, wherein at least one of the delay period and a defined number of information bits in the information packet is controlled so as to cause later-transferred portions of the file to be delayed by increasing amounts.

12. (Previously presented) The system of Claim 11, wherein the application further comprises instructions to perform the step of increasing the defined delay period after each iteration of the repeating step.

13. (Previously presented) The system of Claim 11, wherein the application further comprises instructions to perform the step of setting the defined delay period to a selected predetermined value after each execution of the pausing step.

14. (Previously presented) The system of Claim 11, wherein the application further comprises instructions to perform the step of initializing the defined delay period to a calculated value prior to the removing step.

15. (Previously presented) The system of Claim 11, wherein the application further comprises instructions to perform the step of initializing the defined delay period to a selected predetermined value prior to the removing step.

16. (Previously presented) The system of Claim 11, wherein the application further comprises instructions to perform the step of setting the defined delay period to a calculated value after each execution of the pausing step.

17. (Previously presented) The system of Claim 16, wherein the application further comprises instructions to perform the step of determining the calculated value from the file size or the file type.

18. (Previously presented) The system of Claim 11, wherein the application further comprises instructions to perform the step of setting the defined number of information bits in the packet of information to a calculated value after each execution of the pausing step.

19. (Previously presented) The system of Claim 11, wherein the application further comprises instructions to perform the step of setting the defined number of information bits in the packet of information to a selected predetermined value after each execution of the pausing step.

20. (Previously presented) The system of Claim 11, wherein the application further comprises instructions to perform the step of initializing the defined number of information bits in the packet of information prior to said removing step.

APPENDIX B

EVIDENCE

NONE.

APPENDIX C

RELATED APPEALS AND INTERFERENCES

NONE.